

---

# Spatio-Temporal Image Boundary Extrapolation

---

**Apratim Bhattacharyya**  
Max Planck Institute for Informatics  
Saarbrücken, Germany  
abhattach@mpi-inf.mpg.de

**Mateusz Malinowski**  
Max Planck Institute for Informatics  
Saarbrücken, Germany  
mmalinow@mpi-inf.mpg.de

**Mario Fritz**  
Max Planck Institute for Informatics  
Saarbrücken, Germany  
mfritz@mpi-inf.mpg.de

## Abstract

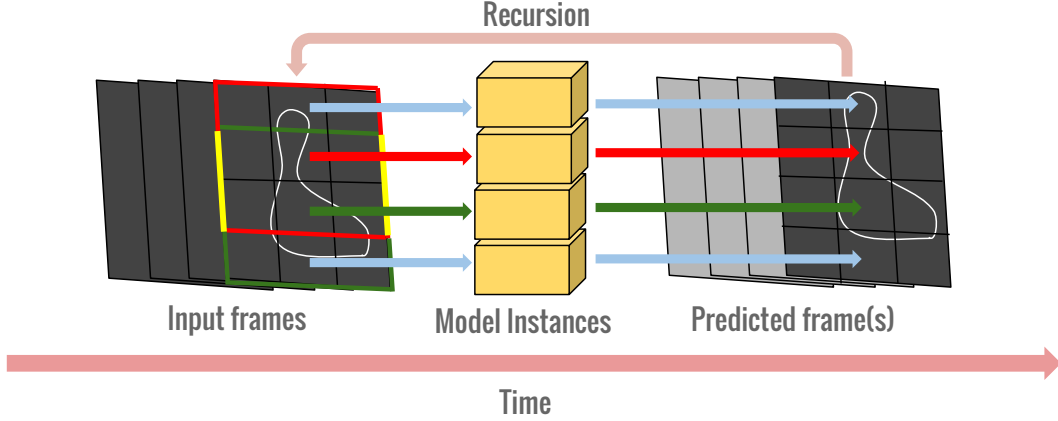
Boundary prediction in images as well as video has been a very active topic of research and organizing visual information into boundaries and segments is believed to be a corner stone of visual perception. While prior work has focused on predicting boundaries for observed frames, our work aims at predicting boundaries of future unobserved frames. This requires our model to learn about the fate of boundaries and extrapolate motion patterns. We experiment on established real-world video segmentation dataset, which provides a testbed for this new task. We show for the first time spatio-temporal boundary extrapolation in this challenging scenario. Furthermore, we show long-term prediction of boundaries in situations where the motion is governed by the laws of physics. We successfully predict boundaries in a billiard scenario without any assumptions of a strong parametric model or any object notion. We argue that our model has with minimalistic model assumptions derived a notion of “intuitive physics” that can be applied to novel scenes.

## 1 Introduction

Humans possess the skill to imagine future states of an observed scene. Part of this skill is footed on an intuitive understanding of physics [1, 2]. Observing a moving ball, we have a reasonably good estimate about the future trajectory of the ball, which lets e.g. a goal keeper catch the ball. Similarly, on a game of billiard we have to choose our action under the prediction of future states of the table in order to win the game. These skills are key to interacting with such dynamic objects that have deterministic fate and let humans excel at such complicated tasks.

Most interestingly, it has been argued that we perform this type of predictions without necessarily using an explicit or formal understanding of the physics which gives rise to these phenomenon. Humans seem to acquire this particular type of physical understanding in a data-driven, learning-based approach from prior experiences [3, 4, 5].

Recently, there has been an increased interest in modeling and predicting phenomena that are governed by the laws of physics [6, 7, 8, 9]. These models typical are parametric of some sort or only predict a qualitative outcome of the scene. More generally, full future frame predication has been studied that is agnostic to the underlying cause of the change depicted in the sequence [10]. In contrast to the physical models, only very short range predictions have been shown and there are blurriness problems with predicting the full future appearance. [11] has predicted future segmentation masks, but also suffers from blurriness in the prediction.



**Figure 1:** General framework that encapsulates all our architectures.

Recently a lot of progress has been made in the field of video segmentation supported by datasets like VSB100 [12]. The performance of any video segmentation algorithm is measured with respect to a set of human annotations. Humans tend to annotate semantically coherent objects and regions as segments. The segmented natural videos discard many details of natural videos mentioned above which are hard for a model to learn and still captures the important objects as boundaries. The boundaries between segments gives rise to boundary images.

Our main contribution is the first model to predict future boundary frames of segmented videos and to explore the performance of these models to under structured motion – physical and non-physical ones. We evaluate performance both on real world videos from VSB100 and synthetic videos. Moreover, we show that our models can develop an intuitive understanding of physics from raw visual input without any strong parametric model of the motion or “object notion”. That is, the model does not know a priori the location or type of the objects it is supposed to model.

## 2 Related work

**Frame prediction.** This problem has been recently explored in [10, 13, 14]. The work of Srivastava et al. [13] focused on learning representations of video sequences. They used an LSTM encoder unit to encode videos into a vector which they used for predicting future frames. In Ranzato et al. [10] the authors focused on the problem of blurring caused by using the mean squared loss as an objective function. They sought to remedy this problem by discretizing the input through k-means atoms and predicting on this vocabulary instead. The work of Mathieu et al. [14] also focused on this problem. They proposed using adversarial loss, which lead to improved results over Ranzato et al. [10]. These works have focused on natural videos or datasets like MNIST digits. Also, the ability of the models developed in the previously mentioned works to learn the dynamics of structured motion has not been explored. Works of Sutskever et al. [15], Michalski et al. [16] generate frames of videos of bouncing balls, but their dataset is very limited in size and resolution. Moreover, they do not consider generalization.

**Equation parameter estimation.** Works like that of Wu et al. [6], Mottaghi et al. [8] focus on predicting outcomes of physical events in videos or images. In [6] the authors propose “Galileo” that estimates the physical properties of objects and inverts a physics engine to predict outcomes. While in [8], the authors predict the motion of objects using a single query image using a neural network which matches the image to a moment in a video which is closest in describing the dynamics of motion of the scene depicted in the image.

**Intuitive physics.** The ability of artificial neural networks to develop such an intuitive understanding of physics from raw visual input has been recently explored in [7, 9, 11]. Fragkiadaki et al. [9] had developed a model which could predict futures states of balls moving on a billiard table. Whereas, Lerer et al. [11] and Li et al. [7] had developed a model which could predict the stability of towers made out of blocks. The work of Lerer et al. [11] could also predict future locations of the blocks. However, both Fragkiadaki et al. [9] and Lerer et al. [11] have an “object notion”. Li et al. [7] focuses only on predicting the outcome not the exact state.

**Video segmentation.** Video segmentation as the task of finding consistent spatio-temporal boundaries in a video volume has received significant attention over the last years [12, 17, 18, 19], as it provides an initial analysis and abstraction for further processing. In contrast, our approach aims at extrapolating these boundaries into the future without any video observed for the future frames. Our proposed model could serve as an expectation on boundaries in future frames and therefore help improve video segmentation prediction in future work.

### 3 Models

In order to extrapolate spatio-temporal boundaries to future unobserved frames, we are building on the recent success of deep learning that allows to construct flexible and end-to-end trainable architectures with strong visual feature encoders, predictors and the recent success of formulating decoders which reconstruct a target image. We are facing several key challenges that will be reflected in the design choices of our architecture:

**Large Spatio-Temporal Receptive Field.** The output layer neurons should have a wide receptive field to preserve long range spatial and temporal dependencies and learn about interaction with other boundaries in a spatio-temporal context. Therefore we explore multiple convolutional layers optionally with pooling or with fully connected layers.

**Preserving Resolution and Preventing Blurred Output.** The models must maintain resolution in order to derive a high fidelity output boundary map. Excessive pooling or tight bottlenecks with fully connected layers have been shown successful in classification tasks, but also have shown to induce image degradations for image synthesis tasks [10].

**Varying resolution.** In order to facilitate extrapolation in different scenarios with changing resolutions and aspect ratios, we target a patch based approach. Such an approach has been successfully used by [10, 14]. Moreover, such an approach simplifies the learning problem by reducing the input dimensionality and modelling reoccurring motion patterns locally.

**Extrapolation over long time scales.** In order to make local predication globally consistent, we have to introduce spatio-temporal dependencies. Consider a video of a moving ball. The trajectory of a ball might intersect with multiple patches. To correctly extrapolate the motion far into the future, instances of a model predicting on neighboring patches need to communicate. In order to allow for exchange of information between different patch extrapolators, we will experiment with larger overlapping receptive fields, that constitute a read-write architecture, where the past frames (observed or already extrapolated ones) serve as a kind of shared memory. However, this potentially makes the learning problem more complex.

#### 3.1 Prediction modes

Here we consider two modes for predicting future frames:

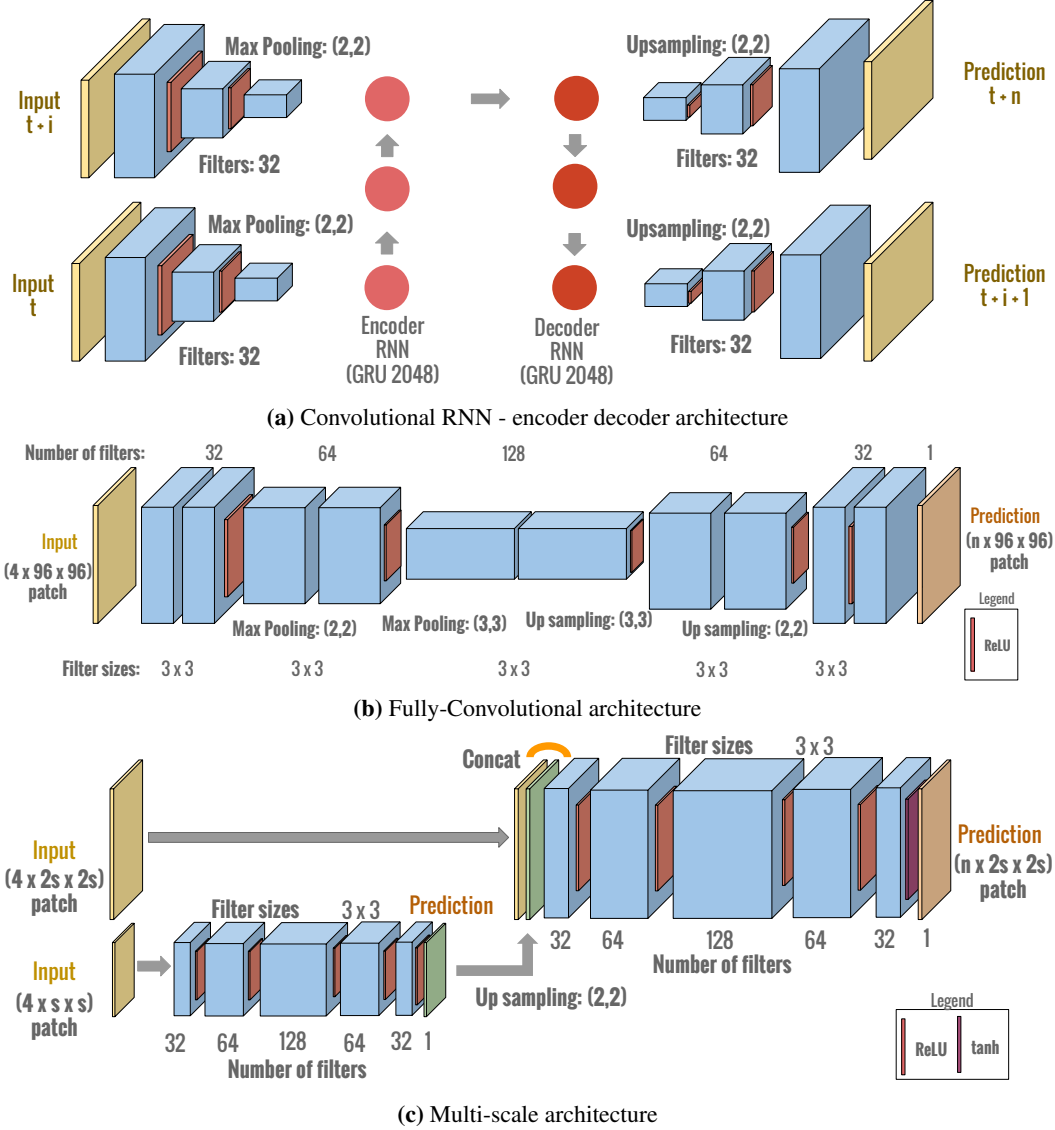
**Sequence to sequence.** Here a model takes a sequence of past frames and predicts another sequence. This mode of prediction has been widely explored in machine translation [20] where the input and output sequences are natural language sentences. Srivastava et al. [13] applied this framework to sequence of images. However, output sequence length is limited by memory and processing time and communication between the patches during extrapolation becomes impossible.

**Recursive.** A model which predicts only one future frame can be made to predict sequences by sampling from its output and using the output as input in the next time step. This mode allows the instances of the model predicting on neighbouring patches to communicate (neighbouring patches are also given as input to model instance i.e. a context). However, in Mathieu et al. [14] this mode led to better short term but worse long term results. This is understandable as the model does not get to learn the long term dynamics of motion.

#### 3.2 Model architectures

We propose the following architectures for boundary extrapolation while keeping in mind the key challenges outlined previously. They all fit in the general framework outlined in Figure 1 and can operate in both the prediction modes outlined previously.

**RNN - encoder decoder.** This model architecture has been explored in Srivastava et al. [13]. This architecture consists of an encoder LSTM unit which reads in the input frame sequence one time step



**Figure 2:** Our concrete architectures.

at a time and produces a single vector as output. This vector is then read by either a dense layer with a non-linearity to produce the final output in the recursive mode or by a decoder LSTM to produce the output sequence in the sequence to sequence mode. Here, we have used GRU units [21] instead of LSTM units because they provide faster convergence without loss of performance.

**Convolutional RNN - encoder decoder.** Convolutional Neural Networks with convolutional layers which extract high quality location invariant features have been very successful in various tasks such as object recognition [22]. We extend the architecture described previously with convolutional layers to extract features along with ReLU non-linearities [23] and pooling layers to extract features which is then fed into an encoder GRU unit (see Figure 2a). To ensure output resolution we up-sample the output of the GRU unit followed by convolution. This can be thought of as convolution with a fractional stride.

**Fully Convolutional.** Both the previously discussed architectures have output neurons with wide receptive fields but they are autoencoders [13] which involve compression. One way to deal with this problem is to increase the size of the encoder (decoder). Another way is to use end to end convolutions instead. For the output neurons of such a model to have a large perceptive field will require many layers of convolutions, making it computationally expensive. We can try to balance

both these factors by using moderate number of pooling layers and an increasing number of filters (with ReLU) and upsampling followed by convolution to maintain resolution (see Figure 2b).

**Multi-Scale.** Another approach to maintain resolution and preserve long range spatial dependencies is to use a multi-scale architecture akin to a Laplacian pyramid [24]. Such an architecture has been used successfully for generating natural images [24] and predicting future natural frames [14]. We use four scales, each downsampled by a factor of two from the one before. Each level captures image structure present at a particular scale. This creates a wide receptive field for the output layer neurons without pooling. We use a five layer convolutional network at each scale with ReLU non-linearities and a tanh non-linearity at the end. The output at a certain scale is upsampled and used as input at the next larger scale as a candidate future frame (see Figure 2c).

### 3.3 Loss function

A variety of loss functions have been tried for frame prediction [10, 13, 14]. These include L1, L2 (also MSE or Mean Squared Error), Adversarial, sharpness measures like GDL(Gradient Difference Loss) etc. It has been observed that L2 loss leads to blurry predictions on grayscale or RGB data because in case of natural images it is possible to obtain low error by blurring the input. Adversarial loss produces sharper results which look more similar to natural images. However, as we want to predict boundary images this effect should be limited. Thus, as a starting point we use the mean squared error.

## 4 Experiments

We evaluate the models in subsection 3.2 first on real data and then on long range extrapolation on sequences with structured, deterministic motion. We convert each video into 32x32 pixel patches. The network observes a patch along with its eight neighbouring patches (when trained with context) at the current time-step and three time-steps into the past. When we use a context, we let the networks predict the next 96x96 patch(s) and use the middle 32x32 patch as high confidence output and discard the rest. This allows for fair comparison with the multi-scale architecture which has same output resolution as input. We use boundary precision recall (BPR) [12] as the evaluation metric. This metric can be defined for a set  $P$  of predicted boundary images and  $G$  of corresponding ground truth boundary images as:

$$P = \frac{\sum_{B_p \in P, B_g \in G} |B_p \cap B_g|}{\sum_{B_p \in P} |B_p|} \quad R = \frac{\sum_{B_p \in P, B_g \in G} |B_p \cap B_g|}{\sum_{B_g \in G} |B_g|} \quad F = \frac{2PR}{P + R}$$

where  $P$  is boundary precision,  $R$  is boundary recall and  $F$  is the combined F-measure.

### 4.1 Real data

**Training.** We use the VSB100 dataset which contains 100 videos with a of maximum 121 frames each. We randomly choose 30 videos for training and 30 for testing videos from the VSB100 dataset. The videos contains a wide range of objects of different sizes and shapes. The videos also have a wide variety of both object and camera motion. We use the hierarchical video segmentation algorithm in Khoreva et al. [25] to segment these videos. The output is a ultra-metric contour map (ucm). Boundaries higher in the hierarchy represent more semantically coherent entities like animals, vehicles etc are therefore more stable temporally. We discard boundaries belonging to the lowest level of the hierarchy as they are very unstable temporally. However, we keep the rest intact and use their values as a confidence measure.

**Evaluation.** As the predicted boundaries have different confidence values, we threshold the predictions before comparison to the ground-truth. We vary the threshold to obtain a precision-recall curve and report the area under the curve (AUC) along with the best F-measure across all thresholds. We evaluate each model under three prediction modes: **sequence to sequence**, **recursive (with context)** and **recursive (without context)**. We always include the last input frame as a baseline. As many boundaries do not change between frames in the videos of VSB100, this is not a bad baseline especially when we are predicting one step into the future.

**Table 1:** AUC and best F measure on VSB100

Time-steps	Last Input	RNN	Conv-RNN	Fully-Conv	Multi-Scale
AUC: Sequence to Sequence					
t + 1	0.163	0.101	0.153	0.269	<b>0.297</b>
t + 2	0.096	0.072	0.110	0.193	<b>0.200</b>
t + 4	0.060	0.039	0.066	<b>0.116</b>	<b>0.115</b>
Best F-measure: Sequence to Sequence					
t + 1	0.398	0.278	0.323	0.423	<b>0.435</b>
t + 2	0.304	0.240	0.286	0.358	<b>0.361</b>
t + 4	0.242	0.182	0.233	<b>0.284</b>	<b>0.283</b>
AUC: Recursive (Without context)					
t + 1	0.163	0.135	0.209	0.292	<b>0.304</b>
t + 2	0.096	0.068	0.099	0.170	<b>0.184</b>
t + 4	0.060	0.027	0.031	0.084	<b>0.087</b>
Best F-measure: Recursive (Without context)					
t + 1	0.398	0.314	0.367	0.432	<b>0.438</b>
t + 2	0.304	0.229	0.279	0.351	<b>0.356</b>
t + 4	0.242	0.141	0.159	0.255	<b>0.257</b>
AUC: Recursive (With context)					
t + 1	0.163	0.125	0.168	0.286	<b>0.309</b>
t + 2	0.096	0.064	0.067	0.159	<b>0.185</b>
t + 4	0.060	0.023	0.017	0.062	<b>0.087</b>
Best F-measure: Recursive (With context)					
t + 1	0.398	0.304	0.341	0.426	<b>0.439</b>
t + 2	0.304	0.232	0.229	0.341	<b>0.357</b>
t + 4	0.242	0.134	0.097	0.219	<b>0.258</b>

**Discussion of results.** We report the results in Figure 3 and Table 1. Overall, the Multi-Scale architecture (red lines) consistently outperforms the others. The Fully-Convolutional architecture (blue lines) is a close second. Adding convolutional layers (magenta lines) improves the performance of the RNN encoder-decoder architecture (cyan lines), however both have trouble beating the last input baseline.

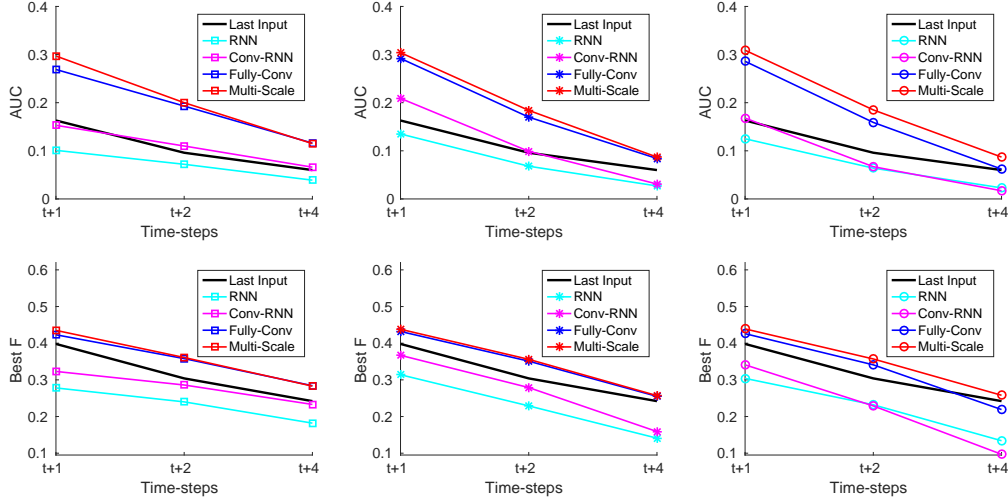
*Sequence to sequence vs Recursive (with and without context):* As expected the sequence to sequence prediction mode performs better than the recursive prediction mode especially at the t + 4 time-step. However, in both modes both precision and recall drop rapidly with time-steps.

*With or Without Context:* Only the Multi-scale architecture is able to deal with high input data dimensionality induced by a context. The others, being simpler benefit from not having a context.

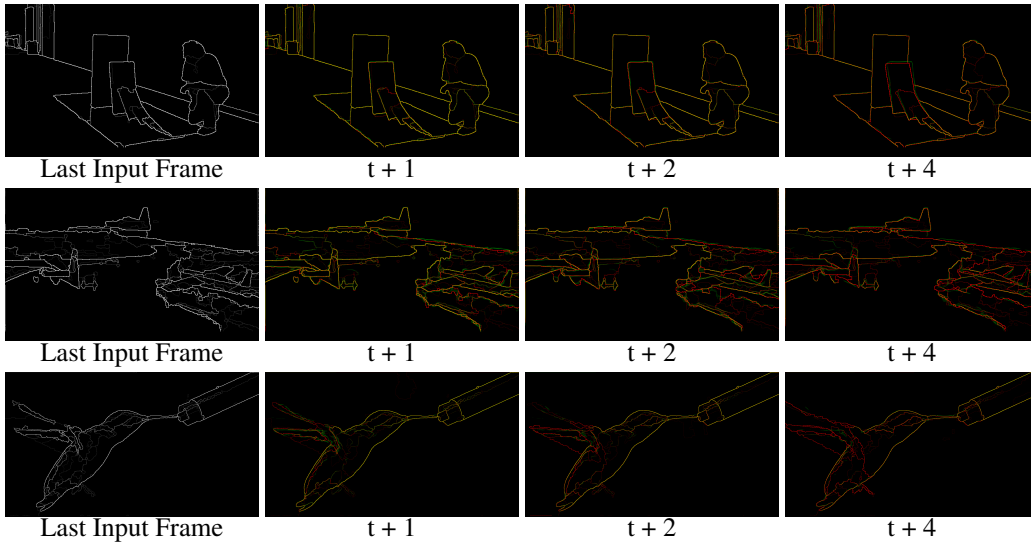
*Qualitative evaluation:* Closer inspection of the results, suggests that the networks are not able to deal with large or unstructured motion. The networks in such situations react by blurring the boundaries, as a consequence of using the mean squared error. While predicting recursively this leads to loss of boundary confidence and eventual disappearance of the boundaries. However, the network is able to predict correctly in case of smooth motion e.g. the predictions in Figure 4 from the videos airplane and dominoes.

## 4.2 Synthetic data

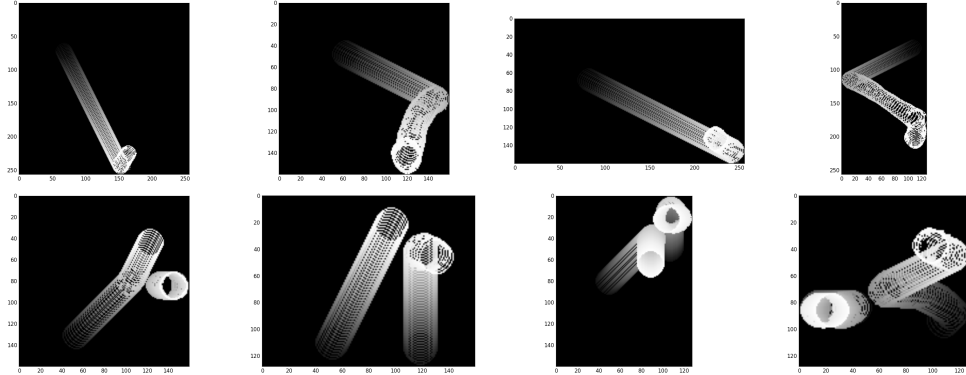
Dynamics of motion in the videos in the VSB100 dataset is frequently very complex and involve difficult to predict and non-deterministic actions of actors. To evaluate the performance of the models on structured, deterministic motion, we test the models trained on videos from the VSB100 on synthetic data. The synthetic data is sampled from worlds which consists balls moving on a frictionless surface with a boundary, akin to a billiard table. We used the pygame module of python



**Figure 3:** Top left to right: AUC for prediction modes: Sequence to Sequence, Recursive (without context), Recursive (with context). Bottom Left to right: Best F-measure for prediction modes: Sequence to Sequence, Recursive (without context), Recursive (with context).



**Figure 4:** Columns top to bottom: Predictions on dominoes, airplane and hummingbird sequences from VSB100. In each predicted frame the actual boundary is coded in the red channel and extrapolated in green channel. Yellow means correct predictions. Extrapolated boundaries are usually sharp especially under smooth motion in dominoes and airplane. In the hummingbird sequence, the models are not able to deal with the fast wing motion, but correctly extrapolates the slower moving legs.



**Figure 5:** Trails produced by super-imposing extrapolated boundaries.

to create such worlds and sample binary boundary images from them. As the target is always a binary image, we report only the best F-measure obtained by varying the output threshold parameter. Moreover, we consider only the recursive mode of prediction (with context) as the sequence to sequence mode and recursive mode without a context is limited by factors mentioned in section 3.

**Data generation from single ball worlds.** We sample 30 sequences from the following parameters to create a synthetic test set:

**Table size:** Side length randomly sampled from  $\{96, 128, 160, 192, 256\}$  pixels.

**Ball velocity:** Randomly sampled from  $\{\{-3, \dots, 3\}, \{-3, \dots, 3\}\}$  pixels.

**Ball size:** Constant, with a radius of 13 pixels.

**Initial Position:** Uniformly over the table surface.

**Table 2:** Best F of models trained on VSB100 on single ball worlds

Time-steps	Last Input	RNN	Conv-RNN	Fully-Conv	Multi-Scale
t + 1	0.141	0.227	0.397	<b>0.612</b>	0.583
t + 2	0.074	0.119	0.255	<b>0.365</b>	0.352
t + 5	0.038	0.010	0.017	0.088	<b>0.097</b>

**Evaluation of models trained on VSB100 on single ball worlds.** We report the performance of the models in Table 2. The models perform remarkably well on this data-set. The models are able to continue the motion of the ball even though they have never seen a moving ball before. However, they do not perform well near collisions and the predicted ball tends to slow down over time.

### 4.3 Extrapolation over long time scales on billiard table worlds

To further test the ability of the models to learn structured motion and extrapolate further into the future, we now train and evaluate the two best performing models from Figure 3 on billiard table worlds.

**Single ball worlds.** We generate a training set using parameters in section 4.2. However, to keep our training set as diverse as possible we prefer short sequences. We restrict each sequence to a maximum length of one or two collisions with walls and set a 50% bias of the initial position of the balls being 40 pixels from the walls. We sample 500 such sequences and train our models on this from scratch. We then test the models on the test set generated in section 4.2 and report the results in Table 3. We include as a baseline a “blind” Fully-Convolutional model, which cannot see the table borders. To beat this baseline, our models need to learn the physics of ball-wall collisions. We see accurate extrapolation even at 20 time-steps in the future.

**Two and three ball worlds.** Worlds with one ball only have collisions with walls. Worlds with more than one ball involve both ball-ball and ball-wall collisions, which make the physics of such



worlds much more complex. To test the ability of the models to predict the future states of such worlds we sample training datasets which contains two and three balls respectively. We use the same parameters as section 4.2, except we limit the scene length to 200 frames. We sample 100 such sequences containing two balls and 50 containing three balls. We use a curriculum learning approach, that is, we initialize the models with the weights learned on single and two ball worlds respectively. We test the models on 30 sequences containing two and three balls respectively. We report the results in Table 3. We include as a baseline the Fully Convolutional models trained on single ball worlds and two ball worlds in the two and three ball world case respectively. This means the model has to learn the physics of ball-ball collisions beat the baselines. This also allows us to evaluate models trained on a simpler world on a more complex one. Again, we see accurate extrapolation even at 20 time-steps in the future.

**Table 3:** Evaluation of models trained on synthetic data (training on the same world)

Time-steps	Last Input	Baseline	Fully-Conv	Multi-Scale
Best F: Evaluation on single ball worlds				
t + 1	0.141	0.964	<b>0.994</b>	0.987
t + 5	0.038	0.791	<b>0.956</b>	0.900
t + 20	0.002	0.637	<b>0.709</b>	0.632
Best F: Evaluation on two ball worlds				
t + 1	0.246	0.941	0.951	<b>0.969</b>
t + 5	0.114	0.776	0.848	<b>0.896</b>
t + 20	0.101	0.545	0.566	<b>0.681</b>
Best F: Evaluation on three ball worlds				
t + 1	0.246	0.950	<b>0.969</b>	<b>0.968</b>
t + 5	0.118	0.823	0.864	<b>0.892</b>
t + 20	0.090	0.550	0.585	<b>0.700</b>

**Extrapolation over very long time scales.** Although we evaluate only 20 timesteps into the future in Table 3, our models are stable over longer time-horizons. We gave the first 4 frames as input and asked the models to extrapolate 100 frames into the future. We superimpose the frames to produce the trails in Figure 5. However, we noticed that sometimes the balls reverse direction mid table and the ball(s) get deformed or disappear altogether.

## 5 Conclusion

We propose a new challenge of predicting boundaries in future video frames as well as several architectures for this novel problem of boundary extrapolation. We investigated a range of design choices and found that our “Multi-Scale” architecture works best on both real and synthetic data. In contrast to prior work, we observe that our model formulation obtains accurate and sharp results even with mean squared error. This lends support to our claim that boundary extrapolation is indeed a better behaved problem than natural frame prediction. Moreover, accurate results on varied scenarios involving billiard balls shows that our models can develop an intuitive notion of physics as well as could lend itself to formulating expectations over future frames in advanced video segmentation methods.

## References

- [1] Barry Smith and Roberto Casati. *Naive Physics: An Essay in Ontology*. Philosophical Psychology, 1994.
- [2] Michael McCloskey. Intuitive physics. *Scientific american*, 1983.
- [3] Renee Baillargeon. How do infants learn about the physical world? *Current Directions in Psychological Science*, 1994.
- [4] R Baillargeon. A model of physical reasoning in infancy. *Advances in infancy research*, 1995.
- [5] Renée Baillargeon. The acquisition of physical knowledge in infancy: A summary in eight lessons. *Blackwell handbook of childhood cognitive development*, 2002.
- [6] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *NIPS*, 2015.
- [7] Wenbin Li, Seyedmajid Azimi, Aleš Leonardis, and Mario Fritz. To fall or not to fall: A visual approach to physical stability prediction. *arXiv preprint arXiv:1604.00066*, 2016.
- [8] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static images. *arXiv preprint arXiv:1511.04048*, 2015.
- [9] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.
- [10] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.
- [11] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*, 2016.
- [12] Fabio Galasso, Naveen Nagaraja, Tatiana Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *CVPR*, pages 3527–3534, 2013.
- [13] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.
- [14] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.
- [15] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *NIPS*, 2009.
- [16] Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent grammar cells". In *NIPS*, 2014.
- [17] Fabio Galasso, Margret Keuper, Thomas Brox, and Bernt Schiele. Spectral graph reduction for efficient image and streaming video segmentation. In *CVPR*, 2014.
- [18] Peter Ochs, Jagannath Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2014.
- [19] Jason Chang, Donglai Wei, and John Fisher. A video representation using temporal superpixels. In *CVPR*, 2013.
- [20] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [21] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [22] Kevin Jarrett, Koray Kavukcuoglu, Marc’Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.
- [23] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.
- [24] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1486–1494, 2015.
- [25] Anna Khoreva, Rodrigo Benenson, Fabio Galasso, Matthias Hein, and Bernt Schiele. Improved image boundaries for better video segmentation. *arXiv preprint arXiv:1605.03718*, 2016.